PATENTTI- JA REKISTERIHALLITUS
NATIONAL BOARD OF PATENTS AND REGISTRATION

Helsinki 4.12.2001

#3

OIPE JC14
APR 1 1 2002
PATENT & TRADEMARK OFFICE

E T U O I K E U S T O D I S T U S
P R I O R I T Y   D O C U M E N T

PATENTTI- JA REKISTERIHALLITUS • PATENT- OCH REGISTERSTYRELSEN

Hakija Applicant	**Nokia Mobile Phones Ltd** **Espoo**
Patenttihakemus nro Patent application no	20010239
Tekemispäivä Filing date	08.02.2001
Kansainvälinen luokka International class	H04L
Keksinnön nimitys Title of invention	

CERTIFIED COPY OF PRIORITY DOCUMENT

"Playback of streamed media"
(Virtaustoistetun median toistaminen)

Hakemus on hakemusdiaariin **03.12.2001** tehdyn merkinnän mukaan siirtynyt **Nokia Corporation** nimiselle yhtiölle, **Helsinki.**

The application has according to an entry made in the register of patent applications on **03.12.2001** been assigned to **Nokia Corporation, Helsinki.**

Täten todistetaan, että oheiset asiakirjat ovat tarkkoja jäljennöksiä patentti- ja rekisterihallitukselle alkuaan annetuista selityksestä, patenttivaatimuksista, tiivistelmästä ja piirustuksista.

This is to certify that the annexed documents are true copies of the description, claims, abstract and drawings originally filed with the Finnish Patent Office.

Pirjo Kalla
Tutkimussihteeri

Maksu	300,-	mk
Fee	300,-	FIM

Maksu perustuu kauppa- ja teollisuusministeriön antamaan asetukseen 1782/1995 Patentti- ja rekisterihallituksen maksullisista suoritteista muutoksineen.

The fee is based on the Decree with amendments of the Ministry of Trade and Industry No. 1782/1995 concerning the chargeable services of the National Board of Patents and Registration of Finland.

L4

Playback of Streamed Media

Field of the Invention

The present invention relates generally to the streaming of media over packet-based networks. More specifically, the invention relates a buffering mechanism for improving playback of the streamed media from packet delay variation due to encoding and packetisation.

Background of the Invention

In conversational packet-switched multimedia systems, e.g., in IP-based video conferencing systems, different types of media are normally carried in separate packets. Moreover, packets are typically carried on top of a best-effort network protocol that cannot guarantee a constant transmission delay, but rather the delay may vary from packet to packet. Consequently, packets having the same presentation (playback) time-stamp may not be received at the same time, and the reception interval of two packets may not be the same as their presentation interval (in terms of time). Thus, in order to maintain playback synchronization between different media types and to maintain the correct playback rate, a multimedia terminal typically buffers received data for a short period (e.g. less than half a second) in order to smooth out delay variation. Herein, this type of buffer is referred to as a delay jitter buffer. In conversational packet-switched multimedia systems buffering can take place before and / or after media data decoding.

Delay jitter buffering is also applied in streaming systems. Due to the fact that streaming is a non-conversational application, the delay jitter buffer required may be considerably larger than in conversational applications. When a streaming player has established a connection to a server and requested a multimedia stream to be downloaded, the server begins to transmit the desired

stream. The player typically does not start playing the stream back immediately, but rather it buffers the incoming data for a certain period, typically a few seconds. Herein, this type of buffering is referred to as initial buffering. Initial buffering provides the ability to smooth out transmission delay variations in a manner similar to that provided by delay jitter buffering in conversational applications. In addition, it may enable the use of link, transport, and/or application layer retransmissions of lost protocol data units (PDUs). Buffering allows the player to decode and play data from the buffer while allowing the possibility for lost PDUs to be retransmitted. If the buffering period is sufficiently long the retransmitted PDUs are received in time to be decoded and played at the scheduled moment.

Initial buffering in streaming clients provides a further advantage that cannot be achieved in conversational systems: it allows the data rate of the media transmitted from the server to vary. In other words, media packets can be temporarily transmitted faster or slower than their playback rate, as long as the receiver buffer does not overflow or underflow. The fluctuation in the data rate may originate from two sources. The first source of fluctuation is the compression efficiency achievable in some media types, such as video, depends on the contents of the source data. Consequently, if a stable quality is desired, the bit-rate of the resulting compressed bit-stream varies. Typically, a stable audio-visual quality is subjectively more pleasing than a varying quality. Thus, initial buffering enables a more pleasing audio-visual quality to be achieved compared with a system without initial buffering, such as a video conferencing system.

Considering the example of video data in more detail, different frames of a video sequence may be represented by very different amounts of data. This results from the use of predictive encoding techniques. Typically, video encoding standards define at least two types of frame. The principal frame types are INTRA or I-frames and INTER or P-frames. An INTRA frame is encoded on the basis of information contained within the image itself, while a P-

frame is encoded with reference to at least one other frame, usually a frame occurring earlier in the video sequence. Due to the significant temporal redundancy between successive frames of a digital video sequence, it is possible to encode an INTER frame with a significantly smaller amount of data than that required to represent an INTRA frame. Thus, INTRA frames are used comparatively infrequently in an encoded video sequence.

Typically an encoded sequence starts with an INTRA frame (as there is no previous frame available to be used as a reference in the construction of an INTER frame). INTRA frames may be inserted into the sequence periodically e.g. at regular intervals, in order to compensate for errors that may accumulate and propagate through a succession of predicted (INTER) frames. INTRA frames are also commonly used at scene cuts where the image content of consecutive frames changes so much that predictive coding does not provide effective data reduction. Thus, a typical encoded video stream generally starts with an INTRA coded frame and comprises a sequence of INTER frames interspersed with occasional INTRA frames, the amount of data required to represent an INTRA frame being several (e.g. 5 - 10) times greater than that required to represent an INTER coded frame. The amount of data required to represent each INTER frame also varies according to the level of similarity / difference with its reference frame and the amount of detail in the image.

This means that the information required to reconstruct a predictively encoded video sequence is not equally distributed amongst the transmitted data packets. In other words, a larger number of data packets is required to carry the data related to an INTRA frame than is required to carry the data for an INTER frame. Furthermore, as the amount of data required to represent consecutive INTER frames also varies depending on image content, the number of data packets required to carry INTER frame data also varies.

A second source of fluctuation occurs when packet losses in fixed IP networks occur in bursts. In order to avoid bursty errors and high peak bit- and packet-

rates, well-designed streaming servers schedule the transmission of packets carefully and packets may not be sent precisely at the rate they are played back at the receiving end. Typically, network servers are implemented in such a way that they try to achieve a constant rate of packet transmission. A server may also adjust the rate of packet transmission in accordance with prevailing network conditions, reducing the packet transmission rate when the network becomes congested and increasing it if network conditions allow, for example. This typically occurs by adjusting the advertised window of the acknowledgement message sent in TCP (transmission control protocol).

Considering this embedded property of network servers, and in connection with the previously described video encoding system, not only is the information required to reconstruct a predictively encoded video sequence unequally distributed between the transmitted data packets, but the data packets themselves may also be transmitted from the server at a varying rate. This means that a decoder in, for example, a receiving client terminal experiences a variable delay in receiving the information that it requires to construct consecutive frames in a video sequence even if the transmission delay through the network is constant. It should be noted that the term client terminal refers to any end-user electronic device such as handheld devices (PDAs), wireless terminals, as well as desktop and laptop computers and set top boxes. This variation in delay, which arises due to encoding, packetisation and packet transmission from a server can be termed an "encoding" or "server-specific" variation delay. It is independent of, or in addition to, delay jitter that arises due to variations in transmission time within the network.

Hence, initial buffering enables the accommodation of fluctuations in transmitted data rate from the aforementioned disadvantages i.e. encoding or server-specific delay variation and network transmission related delay variation. Initial buffering helps to provide a more stable audio-visual quality and to avoid network congestion and packet losses.

Initial buffering may also be performed after decoding of the received media data. This has the disadvantage that the dimensions of the buffer must be relatively large, as the buffering is performed on decoded data. The combined effect of encoding, server-specific and network transmission delay variations also tends to increase the initial buffering requirement.

Furthermore, the encoding of media data and the way in which encoded data is encapsulated into packets and transmitted from a server causes a decoder in a receiving client terminal to experience a variable delay in receiving the information it requires to reconstruct the media data, even if the transmission delay through the network is constant. Thus, a post-decoder buffer does not provide a means of absorbing this form of delay variation prior to decoding.

Summary of the Invention

Briefly described and in accordance with an embodiment and related features of the invention, in a method aspect there is provided a method of playing back streamed media comprised of a plurality of packets transmitted over a network from a source server to a client device wherein the client device includes a decoder for decoding encoded packets, the method is **characterised in that** the client device further includes a pre-decoder buffer having an initial buffering time and a variable buffer size is established for receiving the transmitted data from the source server prior to decoding in the decoder, and wherein the initial buffering time and pre-decoder buffer size are dynamically adapted for improved playback performance by the source server.

In a system aspect of the invention there is provided a system for the transmission and playback of streamed media comprised of a plurality of transmitted packets, the system comprises:

a source server hosting said streamed media;

a packet network serving as a transmission medium for said packets; and

a client device capable of playing back said streamed media wherein said client device comprises:

a pre-decoder buffer for receiving said transmitted packets from said source server via said packet network; and

a decoder for decoding the packets from the pre-decoder buffer.

Brief Description of the Drawings

The invention, together with further objectives and advantages thereof, may best be understood by reference to the following description taken in conjunction with the accompanying drawings in which:

Figure 1 shows a simplified block diagram of a pre-decoder buffer block in the device terminal architecture, in accordance with an embodiment of the invention; and

Figure 2 shows an example of a data flow in a typical network streaming system illustrating the effect of pre-encoder buffering.

Detailed Description of the Invention

Architectural Overview

In accordance with an embodiment of the present invention, a new buffering block is provided in the terminal architecture in order to provide receiver-side buffering. This buffering block is herein referred to as the pre-decoder buffer.

Figure 1 shows a simplified block diagram of a pre-decoder buffer block in the device terminal architecture, in accordance with the embodiment. The "Transport decoder" 100 decapsulates the code-stream from the received data packets (e.g. RTP data packets). The "Source decoder" 120 decodes the code-stream to a raw data format that can be played back. The "Pre-decoder buffer"

110 operates as a temporary storage between transport decoding and source decoding. In the case of a multi-media data stream comprising more than one type of data, a common pre-decoder buffer is advantageously shared between all real-time media types that are transmitted.

Preferably the pre-decoder buffer is provided in addition to a post-decoder buffer located after a decoder in a streaming client. Advantageously, the play-back buffer is provided in order to absorb network transmission delay variations. Additionally, the post-decoder buffer can also absorb decoding-related delay variations. This is particularly advantageous in the case where more than one type of media data is streamed simultaneously. The use of a post-decoder buffer in this situation enables decoding delay variations introduced by different media decoders to be smoothed out.

Buffering Algorithm

A buffering algorithm operating in accordance with the embodiment is provided to buffer received data in a streaming client and to control the encoding and serving of streams from a network-based streaming server. The algorithm assumes that a pre-decoder buffer according to the invention is provided in the streaming client.

There are two primary factors that affect the behavior of the buffering algorithm i.e. the initial buffering time and the minimum pre-decoder buffer size. The initial buffering time typically refers to the time that elapses between the time when a first media data packet is received and the time when the first media sample is played back. The minimum pre-decoder buffer size typically corresponds to the amount of data (e.g. the number of bytes of data) the streaming client is able to store in addition to the buffering that takes place to cope with the transmission delay variation. In other words, a minimum pre-decoder buffer size is defined for zero-delay reliable transmission networks.

The buffering algorithm is similar to the algorithms described in H.263 Annex B (Hypothetical Reference Decoder) and MPEG-4 Visual Annex D (Video buffering verifier). These algorithms define the buffering behavior for video codecs. It should be noted that these algorithms cannot be used to replace the proposed pre-decoder buffering algorithms, because they are applicable to video only. Moreover, the H.263 Hypothetical Reference Decoder does not support initial buffering or storing of multiple (non-B) frames into the buffer. It should also be noted that the proposed pre-decoder buffering algorithm is fully compatible with the previously mentioned video buffering algorithms. In a practical implementation, the pre-decoder buffer and the video decoder buffer can be combined.

In accordance with the embodiment of the invention, a transmitted data packet stream is constrained to comply with the requirements of the pre-decoder buffer which are defined as follows:

1. The pre-decoder buffer is initially empty.
2. Each received data packet is added to the pre-decoder buffer substantially immediately it is received. All protocol headers at the transmission protocol level (e.g. RTP layer) or any lower layer are removed.
3. Data is not removed from the pre-decoder buffer during a period called the initial buffering time that is started when the first data packet is added to the buffer.
4. When the initial buffering time has expired, a playback timer is started.
5. A data chunk is removed from the pre-decoder buffer substantially immediately when the playback timer reaches its scheduled playback time.
6. When the data is carried over a zero-delay reliable transmission network, the occupancy level of the pre-decoder buffer is not permitted to exceed a certain level called the pre-decoder buffer size.

It should be noted that the requirements above describe the operation without intermediate pause requests. Each new play request (after a pause, for example) will follow the same requirements.

Furthermore, the requirements above are based on the assumption of having a zero-delay reliable transmission network. Thus, in a practical implementation, client-side pre-decoder buffering is likely to be combined with network delay jitter buffering. Consequently, the actual pre-decoder buffer size in a streaming client is likely to be larger than the minimum pre-decoder buffer size discussed above, and the actual initial buffering time is also likely to be longer than the initial buffering time discussed above.

Pre-Decoder Buffering

Figure 2 shows an example of a data flow in a typical network streaming system illustrating the effect of pre-encoder buffering. The bars represent media frames or packets, for example, the dark bars are video data packets (e.g. encoded according to ITU-T recommendation H.263) and the light bars are audio data packets (e.g. AMR). The height of the bars represent the size of a frame (or a packet) in bytes. The processing flow runs from the top to the bottom, and time runs from left to right.

Now referring to Figure 2 in more detail, at first, the input data is compressed. As a result, the video stream has a varying frame rate and frame size, and the audio stream has a constant frame rate but a varying frame size. Next the compressed media streams are encapsulated into packets and transmitted to the network. While encapsulating, the server splits large video frames to multiple packets, and combines a number of small audio frames into one packet. The server transmits packets at regular intervals. A constant network transmission delay is assumed, regardless of the packet size or any other factors. Thus, the relative timing of received packets is the same as when they were sent. The received packets are stored in a pre-decoder buffer. After a

certain initial buffering time, frames are retrieved from the buffer, and the frame removal rate is the same as the frame playback rate. The maximum buffer occupancy level determines the minimum pre-decoder buffer size.

Signalling of Pre-Decoder Buffer Characteristics

In order to ensure certain minimum buffering capabilities in streaming clients, the default buffer characteristics have to be defined. As described earlier, the buffer characteristics can be defined primarily by two factors i.e. the initial buffering time and the minimum pre-decoder buffer size. An example for the default value for the initial buffering time is around one second and the default minimum pre-decoder buffer size of about 30720 bytes. It should be noted that these values are only exemplary and they may be varied to achieve suitable performance for the particular type of delays experienced in the network at the time. The suggested default values are based on practical experiments in a generalized environment that are in no way specific but take into account the most commonly occurring packet transmission scenarios.

In order to allow streaming clients to signal and receive streams requiring more demanding buffering capabilities than the default capabilities, signaling based on the SET_PARAMETER method of RTSP [1] is used in the invention. By way of example, the client terminal device can request the server to set either one or both of the following parameters:

1. initialBufferingTimeInMSec (initial buffering time in milliseconds)
2. preDecoderBufferSizeInBytes (minimum pre-decoder buffer size in bytes)

Parameter values smaller than the default values are not allowed, in which case servers that receive lower values will signal a "Bad Request" if such a request arrives. If the transmitted values are greater than or equal to the default values, servers will verify the transmitted packet stream with the passed values

according to the algorithm. The updated values will then be taken into use substantially immediately the request is received.

The present invention contemplates a pre-decoder buffer as a part of a streaming client. The streaming client operates by following a buffering algorithm in which a streaming server verifies that the transmitted data stream complies with the defined buffering algorithm. Additionally, the invention proposes mechanisms for defining and signaling the buffer capabilities of a streaming client to a streaming server. In this way a streaming server can obtain information about the buffering capabilities of a given streaming client and the encoded data/media transmission rate can be allowed to vary within the limits of the receiver-side pre-decoder buffer. It should be noted that a buffering verifier in a server can be used to ensure that the transmitted packet stream complies with the receiver buffering capabilities. This can be done, for example, by adjusting the transmission times of packets from the server so that the buffering capabilities of the client's pre-decoder buffer are not exceeded. Alternatively, the server may adjust the way in which media data is encoded and packetised. In practice, the buffering verifier can be a buffer running within the server after the transport encoding.

Although the invention has been described in some respects with reference to a specified embodiment thereof, variations and modifications will become apparent to those skilled in the art. It is therefore the intention that the following claims not be given a restrictive interpretation but should be viewed to encompass variations and modifications that are derived from the inventive subject matter disclosed.

CLAIMS

1. A method of playing back streamed media comprised of a plurality of packets transmitted over a network from a source server to a client device wherein the client device includes a decoder for decoding encoded packets, the method is **characterised in that** the client device further includes a pre-decoder buffer having an initial buffering time and a variable buffer size is implemented for receiving the transmitted data from the source server prior to decoding in the decoder, and wherein the initial buffering time and pre-decoder buffer size are dynamically adapted for improved playback performance by the source server.

2. A method according to claim 1 **characterised in that** the client device may submit a request to the source server to set either one or both of the initial buffering time and pre-decoder buffer size.

3. A method according to claim 2 **characterised in that** the initial buffering time and pre-decoder buffer size are dynamically adapted by a buffering algorithm.

4. A method according to claim 2 **characterised in that** a post-decoder buffer is implemented to reduce decoding-related delay variations.

5. A method according to claim 1 **characterised in that** the streamed media is transmitted to a wireless client devices via a wireless data packet network such as GPRS (General Packet Radio Service) or UMTS (Universal Mobile Telecommunications System).

6. A system for the transmission and playback of streamed media comprised of a plurality of transmitted packets, the system comprises:

a source server hosting said streamed media;

a packet network serving as a transmission medium for said packets; and

a client device capable of playing back said streamed media wherein said client device comprises:

a pre-decoder buffer for receiving said transmitted packets from said source server via said packet network; and

a decoder for decoding the packets from the pre-decoder buffer.

7. A system according to claim 6 wherein the packet network is a wireless system such as UMTS or a data packet network such as GPRS.

8. A system according to claim 7 wherein the client device is a wireless terminal compatible for data packet use by said wireless system.

9. A system according to claim 6 wherein the pre-decoder buffer is a variable size buffer.

10. A system according to claim 6 wherein a buffering verifier is implemented in the source server for ensuring that the packets are transmitted at a rate that complies with the buffering capabilities of the client device.

L6

/

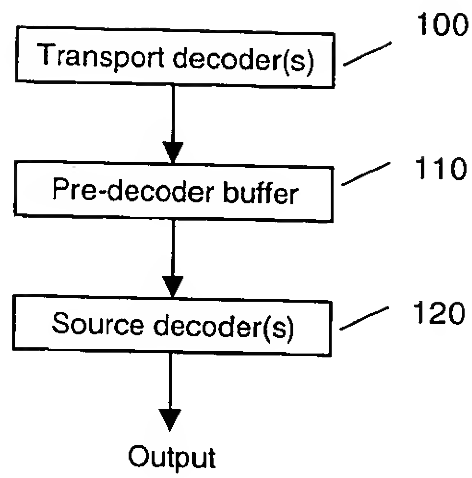
(57) ABSTRACT

The invention discloses a method of improving the playback of streamed media on a client device by overcoming problems caused by variations in the transmission delay of packets due to network and transport protocol operation and variations in encoding/server specific delays. In an embodiment of the invention, a client device has a decoder 120 and a pre-decoder buffer 110 which receives streamed packets from source server via a packet network. The pre-decoder buffer is variable in size and has a variable initial buffering time for receiving the transmitted packets from the source server prior to decoding in the decoder. The initial buffering time and pre-decoder buffer size can be dynamically adapted for improved playback performance by the source server. In a further aspect of the invention, a post-decoder buffer operates in conjunction with the pre-encoder buffer to reduce decoding-related delay variations.

Figure 1

L7

/



Transport decoder(s)
100
Pre-decoder buffer
110
Source decoder(s)
120
Output

Figure 1

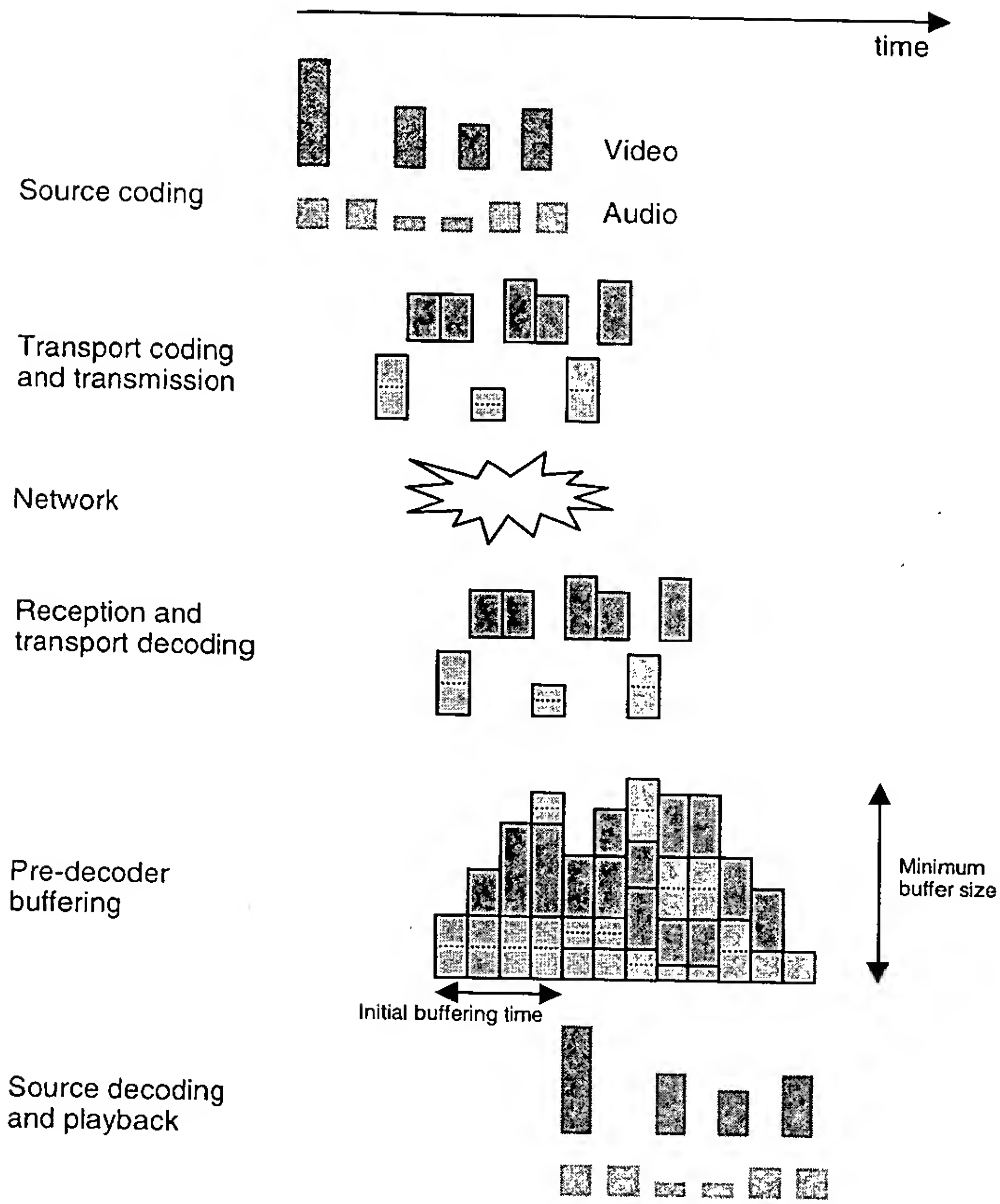

time
Video
Source coding
Audio
Transport coding
and transmission
Network
Reception and
transport decoding
Pre-decoder
buffering
Minimum
buffer size
Initial buffering time
Source decoding
and playback

Figure 2